

### **Remarks**

This REQUEST FOR CONTINUED EXAMINATION and REPLY is in response to the Office Action mailed December 12, 2008, the Advisory Action mailed March 5, 2009, and the Examiner Interview Summary mailed April 9, 2009. A Petition for Extension of Time is submitted herewith, together with the appropriate fee.

This amendment supersedes that filed by Applicant on March 23, 2009. Applicant respectfully submits that the paper filed on March 23, 2009 was intended to be unofficial, for interview purposes only, and requests that the instant paper be considered instead.

#### **I. Interview Summary**

Applicant thanks Examiner Ted Vo for the courtesy of an interview with Adam Hipp on March 22, 2009, during the course of the interview the participants generally discussed the pending claims and the cited references. No claim amendments were proposed during the interview, and no agreement was reached between the parties, as set forth in full in the Examiner Interview Summary mailed April 9, 2009. The Examiner indicated that further review of the application would follow upon filing of the instant Reply.

#### **II. Summary of Examiner's Rejections**

In the Office Action mailed December 12, 2008, Claims 50-63 and 71-75 were rejected under 35 U.S.C. 103(a) as being unpatentable over "Event-Oriented Dynamic Adaptation of Workflows: Model, Architecture, and Implementation" (hereinafter Muller) in view of "Web Service Orchestration" (HP, 1-2003, hereinafter Peltz).

#### **III. Summary of Applicant's Amendments**

The present Reply amends Claims 50, 57 and 75; and adds Claims 76-81, leaving for the Examiner's present consideration Claims 50-63 and 71-81.

#### **IV. Claim Rejections under 35 U.S.C. §103(a)**

In the Office Action, Claims 50-63 and 71-75 were rejected under 35 U.S.C. 103(a) as being unpatentable over Muller in view of Peltz.

#### **Claim 50**

Claim 50 has been amended to more clearly define the embodiment therein. As amended, Claim 50 defines:

50. *(Currently Amended) A system for utilizing a workflow language, comprising:*  
*a computer including a processing device and a workflow compiler operating thereon;*  
*a program source file stored on a computer readable medium, wherein the program source file includes a source code and classes therein;*  
*a workflow definition created using a workflow language that is specified in the form of annotations to the source code and the classes, wherein the annotations to the source code, when compiled by the workflow compiler, are used to define workflow activities at the computer in accordance with the workflow definition, and said workflow language extends the source code with a plurality of workflow constructs, including constructs for defining parallel processing of a workflow and separate workflow branches therein, and wherein the workflow definition further includes a construct to terminate the parallel processing of the workflow when certain conditions are met; and*  
*object code executed by the processing device, the object code configured to*  
*use a workflow program according to said workflow definition, including processing, the plurality of workflow constructs as defined by the annotations to the source code, to activate a workflow, including creating separate workflow processes corresponding to the separate workflow branches,*  
*activate each of the separate workflow processes, in accordance to the annotations to the source code, to subsequently generate activities at the computer as defined by each workflow branch, and*  
*determine when the certain conditions specified in the source file have occurred and then terminating the parallel processing of the workflow.*

As currently amended, Claim 50 defines using a workflow language in the form of annotations to the source code and the classes to enable developers that are used to working with a particular source code to add workflow definitions in the form of annotations to the source code that are used to define workflow activities at the computer in accordance with the workflow definition.

Applicant respectfully submits that, in a traditional system, annotation to source code and the classes would normally be ignored when the source code is processed, because annotations normally are not part of the program itself, and they normally have no direct effect on the operation of the source code they annotate. However, as defined by Claim 50 as amended, in the embodiment therein, such annotations are read by the system and are used to define workflow activities at the computer in accordance with the workflow definition, rather than being ignored.

Muller discloses an agent-based workflow management system that can handle various failure situations. As disclosed therein, the AGENTWORK system supports the definition, the execution and, as its main contribution, the event-oriented and semi-automated dynamic adaption of workflows (page 1). Figure 1-2 of Muller (at page 5) illustrates a simplified architecture of a workflow management system, which includes a workflow definition.

Claim 50, as amended, defines *annotations to the source code and the classes, wherein the annotations to the source code, when compiled by a workflow compiler, define workflow activities at the*

*computer in accordance with the workflow definition.* In the Office Action, it was acknowledged that Muller does not explicitly address that the program source file includes a source code and classes therein, and a workflow definition created using a workflow language that is specified in the form of annotations to the source code and classes. However, it was asserted that Peltz teaches this deficiency of Muller because Peltz discloses that a workflow definition is invoked using WSFL, where WSFL includes Java, i.e., program source file that includes source code and classes therein, and JavaDoc annotation tags in the workflow language that is specified in the form of annotations to the source code and classes (Office Action, page 6). Applicant respectfully traverses this assertion.

In particular, it appears that Peltz provides an overview of different web service technologies but does not disclose that WSFL includes Java. Peltz, at page 5, discloses that WSFL was an IBM proposal to describe both public and private process flows and defines a specific order of activities and data exchanges for a particular process. Applicant submits that in the description of WSFL, Peltz does not disclose or suggest that that WSFL includes Java.

Peltz, at page 11, also discloses WebLogic Workshop technology. As disclosed therein, certain process logic can be written using Java program source code, which includes program source code and classes. Applicant respectfully submits that WSFL is not disclosed in the description of WebLogic Workshop technology.

Thus, while at page 5 Peltz discloses that WSFL was an IBM proposal to describe both public and private process flow; and at page 11 that process logic in WebLogic Workshop technology can be written in Java; Applicant respectfully submits that these are different technologies, and that the discussion of these technologies alone, or in combination, does not disclose or suggest that WSFL includes Java.

Additionally, Applicant respectfully submits that Peltz does not disclose *a workflow language that is specified in the form of annotations to the source code and the classes.* The annotations as defined in Claim 50, as amended, are *used to define workflow activities at the computer in accordance with the workflow definition.* Peltz instead discloses using JavaDoc annotation tags to begin a conversation. As disclosed therein, to begin a conversation, a developer can include the tag “@jws:conversation: phase=start” in the source code (page 10).

Applicant respectfully submits that the use of a JavaDoc annotation tag to start a conversation appears merely to be a way to identify a two-way communication, e.g., between one client application and one Web Service so that messages are always returned to the correct client and that the use of annotations in Peltz are simply a tagging mechanism (page 10).

In contrast, a workflow is a set of steps or actions that occur in a particular order, and in response to specific conditions, in order to perform a task. Applicant submits that this is different from annotations that are used to manage a conversation through simple Java tags, since the Java tags do not perform a set

of steps or actions that occur in a particular order, and in response to specific conditions, in order to perform a task. Accordingly, the annotations in Peltz to begin a conversation is different from the embodiment defined by Claim 50, as amended, which defines *a workflow language that is specified in the form of annotations to the source code and the classes*, where the annotations to source code and classes are used to define workflow activities at the computer in accordance with the workflow definition.

Claim 50 further defines that the *workflow language extends the source code with a plurality of workflow constructs, including constructs for defining parallel processing of a workflow and separate workflow branches therein*. These activities include workflow constructs *for defining parallel processing of a workflow and separate workflow branches therein*. Applicant respectfully submits that the combination of Muller in view of Peltz does not disclose this feature of Claim 50.

In particular, even though Peltz provides an example of using annotations to identify a two-way communication, the use of annotations in Peltz does not disclose that annotations can be used to extend source code with a plurality of workflow constructs. As defined in Claim 50, these constructs *activate each of the separate workflow processes, in accordance to the annotations to the source code, to subsequently generate activities at the computer as defined by each workflow branch*. As described above, the use of annotations in Peltz is a simple tagging mechanism used to manage a conversation and appear to have no direct effect on the operation of the source code they annotate. Accordingly, it is not obvious to combine Peltz with Muller because Peltz does not disclose that *a workflow language that is specified in the form of annotations to the source code and the classes*.

In view of these comments, Applicant respectfully submits that Claim 50, as amended, is neither anticipated by, nor obvious in view of the cited references when considered alone or in combination, and reconsideration thereof is respectfully requested.

#### **Claim 57 and 75**

The comments provided above with respect to Claim 50 are hereby incorporated by reference. Claims 57 and 75, while independently patentable, recite limitations that, similarly to those described above with respect to Claim 50, are not rendered obvious by the cited references. For similar reasons as provided above with respect to Claim 50, Applicant respectfully submits that Claims 57 and 75 are likewise neither anticipated by, nor obvious in view of the cited references, when considered alone or in combination, and reconsideration thereof is respectfully requested.

#### **Claims 51-56 and 58-63**

Claims 51-56, 58-63 and 71-74 depend from and include all of the features of Claims 50 or 57. Claims 51-56, 58-63 and 71-74 are not addressed separately but it is respectfully submitted that these

claims are allowable as depending from an allowable independent claim, and the comments provided above. Reconsideration thereof is respectfully requested.

**V. Claims 76-81**

Claims 76-81 have been newly added by the current Reply. Applicant respectfully submits that no new matter is being added, and that support for Claims 76-81 can be found in the present application as originally filed. Claims 76-81 are not addressed separately but it is respectfully submitted that these claims are allowable as depending from an allowable independent claim, and the comments provided above. Consideration thereof is respectfully requested.

**VI. Request for Interview**

Applicant respectfully submits that the claims, as amended, are now allowable and a notice of allowance is respectfully requested. However, in the event that the next Office Action is not an allowance, Applicant hereby respectfully requests a telephonic interview with the Examiner prior to issuance of the next Office Action in order to expedite the prosecution of this application.

**VII. Conclusion**

In view of the above amendments and remarks, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

Enclosed is a PETITION FOR EXTENSION OF TIME UNDER 37 C.F.R. §1.136 for extending the time to respond up to and including April 20, 2009.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this reply, including any fee for extension of time which may be required.

Respectfully submitted,

Date: April 20, 2009

By: /Adam T. Hipp/  
Adam T. Hipp  
Reg. No. 60,334

Customer No. 23910  
FLIESLER MEYER LLP  
650 California Street, 14th Floor  
San Francisco, California 94108  
Telephone: (415) 362-3800